Numerical multi-loop calculations with the program SecDec

Borowka, Sophia ; Heinrich, Gudrun

Abstract: SecDec is a program which can be used for the evaluation of parametric integrals, in particular multi-loop integrals. For a given set of propagators defining the graph, the program constructs the graph polynomials, factorizes the endpoint singularities, and finally produces a Laurent series in the dimensional regularization parameter, whose coefficients are evaluated numerically. In this talk we discuss various features of the program, which extend the range of applicability. We also present a recent phenomenological example of an application entering the momentum dependent two-loop corrections to neutral Higgs boson masses in the MSSM.

DOI: https://doi.org/10.1088/1742-6596/608/1/012062

Numerical multi-loop calculations with the program SecDec

Sophia Borowka[1], **Gudrun Heinrich**[2,3]

[1] Institute for Physics, University of Zurich, Winterthurerstr.190, 8057 Zurich, Switzerland
[2] Max Planck Institute for Physics, Föhringer Ring 6, 80805 Munich, Germany

E-mail: sborowka@physik.uzh.ch, gudrun@mpp.mpg.de

Abstract. SecDec is a program which can be used for the evaluation of parametric integrals, in particular multi-loop integrals. For a given set of propagators defining the graph, the program constructs the graph polynomials, factorises the endpoint singularities, and finally produces a Laurent series in the dimensional regularization parameter, whose coefficients are evaluated numerically. In this talk we discuss various features of the program, which extend the range of applicability. We also present a recent phenomenological example of an application entering the momentum dependent two-loop corrections to neutral Higgs boson masses in the MSSM.

1. Introduction

In view of the absence of "smoking gun" signals of new physics at the LHC so far, the importance of precision calculations cannot be over-emphasized. This means that higher oder corrections in both the QCD and the electroweak sector need to be evaluated, ideally without neglecting mass effects, and including matched parton showers as far as possible.

These calculations have many facets, however most of them have in common that they involve multi-dimensional integrations over some parameters, for example Feynman parameters in the case of (multi-)loop integrals, or parameters related to the integration over a factorized phase space of subtraction terms for infrared-divergent real radiation. Usually, these calculations are performed within the framework of dimensional regularization, and one of the challenges is to factorise the poles in the regulator ε.

The program SecDec [1, 2, 3] is designed to perform this task, and to integrate the coefficients of the resulting Laurent series in ε numerically, based on the sector decomposition algorithm described in [4, 5]. Other public implementations of sector decomposition can be found in [6, 7, 8, 9].

2. The program SecDec

2.1. Basic structure

The program consists of two main parts, one being designed for loop integrals, to be found in a directory called `loop`, the other one for more general parametric integrals, located in a directory called `general`. The procedure to isolate the poles in the regulator ε and to do the subtractions and integrations is very similar in the two branches. However, only the `loop` part contains the

[3] Speaker; presented at the conference ACAT 2014, Prague, Czech Republic, September 2014.

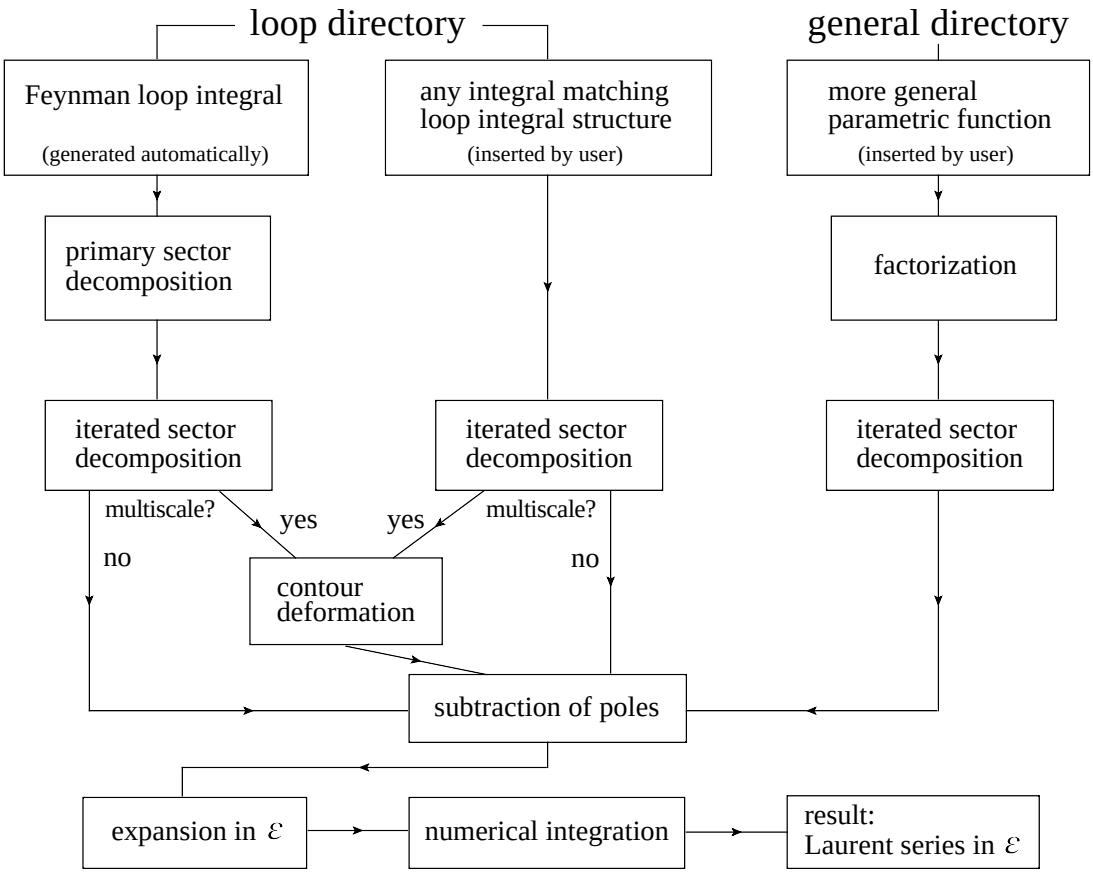

Figure 1. Flowchart showing the main steps the program performs to produce the numerical result as a Laurent series in ϵ.

possibility of contour deformation, because only for loop integrals the analytic continuation can be performed in an automated way, following Feynman's "$i\varepsilon$" prescription. The basic flowchart of the program is shown in Fig. 1. More details about the central column will be given in section 2.5.

2.2. Installation and usage

Installation

The program can be downloaded from `http://secdec.hepforge.org`. Unpacking the tar archive via *tar xzvf SecDec.tar.gz* will create a directory called `SecDec`. Running *./install* in the `SecDec` directory will install the package. Prerequisites are Mathematica, version 6 or above, perl (installed by default on most Unix/Linux systems), a C++ compiler, and a Fortran compiler if the Fortran option is used. Contour deformation is only available in C++. The libraries CUBA [10, 11] and BASES [12], which are used for the numerical integration, come with the package SecDec and will be compiled at the installation stage.

Usage

The user should edit two files (where templates are provided): a text file where parameters like the name of the graph, the order to be expanded in ε or numerical integration parameters are set, and a file in Mathematica syntax, where the graph is defined, optionally either in terms

of propagators or in terms of lines connecting numbered vertices. The template to define the parameters for standard loop integrals is called `paramloop.input`, and the one in Mathematica syntax is called `templateloop.m`. It is recommended that the user copies these template files and renames them before editing them. We will call the edited files `myparamfile.input` and `mytemplatefile.m` in the following. Examples for a number of specific graphs or input functions can be found in the subdirectories `demos` of both the `loop` and the `general` directory. After having edited the input files, SecDec is called as follows:

(i) from the `loop` (or `general`) directory, execute the following command in the shell:
./launch -p myparamfile.input -t mytemplatefile.m
If your files `myparamfile.input, mytemplatefile.m` are in a different directory, say, *myworkingdir*, use the option *-d myworkingdir*, i.e. the full command then looks like *./launch -d myworkingdir -p myparamfile.input -t mytemplatefile.m*, executed from the directory `SecDec/loop` or `SecDec/general`.
If the option *-p myparamfile.input* is omitted, the file `paramloop.input` will be taken as default. Likewise, if the option *-t mytemplatefile.m* is omitted, the file `templateloop.m` will be taken as default.

(ii) Collect the results. If the calculations are done sequentially on a single machine, the results will be collected automatically. If the jobs have been submitted to a cluster, when all jobs have finished, use the command *perl results.pl [-d myworkingdir -p myparamfile]*. In both cases, the files containing the final results will be located in the `graph` subdirectory specified in the input file.

2.3. Topology definition

In the Mathematica input file `mytemplatefile.m`, the graph can be defined in two ways: either by labeling the vertices and listing the connections between vertices (corresponding to the flag `cutconstruct=1`), or by specifying the momentum flow explicitly (corresponding to `cutconstruct=0`). The program will then construct the integrand in terms of Feynman parameters automatically. For tensor integrals, i.e. integrals with loop momenta in the numerator, the propagators have to be given in terms of momenta, i.e. `cutconstruct=0` must be used, as tensor integrals are not shift invariant.

An example for the construction of the graph polynomials based on labelled vertices is given in subsection 3.1.

2.4. Controlling the different stages of the calculation

As the program consists of a purely algebraic part and a numerical part, it can be useful to perform the calculation only up to a certain level, instead of launching immediately the full program chain including the numerical integration. For example, to get an idea about the pole structure of an integral, one can first perform only the iterated sector decomposition to factor out the parameters exhibiting the poles in ε to expect. This can be achieved by setting `exeflag=0` in `myparamfile.input`. The following stages can be selected with the `exeflag`:

- 0: The iterated sector decomposition is done. The scripts to do the subtractions, the expansion in epsilon and to launch the numerical part are created (scripts batch* in the subdirectory *graph*), but not run.
- 1: The subtraction and epsilon expansion is performed and the resulting functions are written to Fortran/C++ files.
- 2: All the files needed for the numerical integration are created.
- 3: The compilation of the Fortran/C++ files is launched to make the executables.
- 4: The executables are run to perform the numerical integration.

2.5. Evaluation of user-defined functions with arbitrary kinematics

In the standard setup, the program will construct the graph polynomials $\mathcal{F}$ and $\mathcal{U}$ and then proceed directly to the so-called *primary sector decomposition* [4, 5], which serves to integrate out the constraint $\delta(1 - \sum_i x_i)$ in a way which preserves the property that all the endpoint singularities are located at the origin of parameter space, rather than creating singularities at $x_i = 1$.

However, there are certain situations where it is useful to start the iterated decomposition at a later stage, where the delta-function already has been integrated out analytically, or for an integrand which does not contain such a delta-function at all. If one can find a convenient parametrisation and integrate out one Feynman parameter analytically, this can be beneficial for complicated integrals, because it reduces the number of integration variables for the subsequent Monte Carlo integration and therefore will improve the numerical efficiency.

In such cases, the user can skip the primary sector decomposition step and insert the functions to be factorized directly into the Mathematica input file. The purpose of this option is to be more flexible with regards to the functions to be integrated, such that expressions for loop integrals which are not in the "standard form" can be dealt with as well. This includes the possibility to perform a deformation of the integration contour into the complex plane, taking the user-defined functions as a starting point. Oriented at the functions $\mathcal{F}$ and $\mathcal{U}$ for the "standard" loop case, the user-defined functions can encompass the product of two arbitrary polynomial functions with different exponents, and an additional numerator.

The command to launch the calculation is the same as for the standard setup, except that the Mathematica input file looks different (for an example see the file `templateuserdefined.m`), and that the extension `-u` should be appended:
./launch -p myparamfile.input -t mytemplateuserdefined.m -u
The "-u" stands for "user defined" and skips the primary sector decomposition step.

2.6. Scanning over ranges of numerical parameters

The algebraic part of SecDec can deal with symbolic expressions for the kinematic invariants or other parameters contained in the integrand. Therefore, the decomposition and subtraction parts only need to be done once and for all, and then can be employed for the calculation of many different numerical points. The program comes with scripts which facilitate the scanning over ranges of numerical values for the kinematic invariants. There is a perl script `helpmulti.pl` which can be used to produce an input file `multiparam.input` containing ranges of numerical values for the invariants/symbolic parameters contained in the integrand. This way the user does not have to type in all the numerical values by hand. Each line in `multiparam.input` defines a new run. An example of a `multiparam.input` file is contained in both the `loop` and the `general` directories. A detailed description of this option is also given in [1].

In order to launch the calculations for the values specified in `multiparam.input`, the command
perl multinumerics.pl [-d myworkingdir] -p multiparamfile
should be issued.

Please note that before executing the script `multinumerics.pl`, the Mathematica-generated functions must already be existent. The simplest way to do this is to make one run with *exeflag=1* in the single-run parameter file.

To collect the results: in single-machine mode (*clusterflag=0*), the results will be collated automatically and written to the `graph` directory specified in the parameter file. In *cluster* mode the *multinumerics.pl* script has to be run again with the argument '1' appended i.e. *perl multinumerics.pl [-d myworkingdir] -p multiparamfile 1* to collect the results. In both cases files with the extension `[pointname][i].gpdat` will be created for each coefficient of ε^i in the Laurent expansion, which can be plotted easily. For example, if the parameter s has been scanned over,

and s is the first element in the list of invariants, setting `xplot=1` in `multiparam.input` will define s to be on the x-axis, and therefore s will be printed into the first column of the `.gpdat` file. Thus, the columns in these result files will be
`(s) (real part) (error_real_part) (imaginary part) (error_imaginary_part) (timings)`.
Result files for each individual point (extension `.res`) will also be created.

3. Applications

3.1. Two-loop four-point integrals with two mass scales

Here we give an example for the definition of a non-planar two-loop four-point function, containing both internal masses, and massive external legs with a different mass.

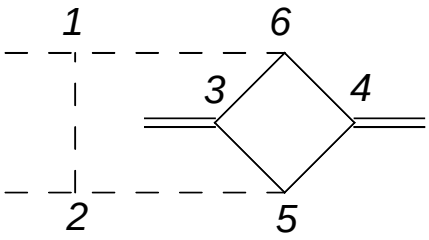

Figure 2. Example of a non-planar two-loop box diagram with two mass scales. Dashed lines denote massless propagators, solid lines massive propagators. The double lines denote massive legs with a mass different from the internal mass.

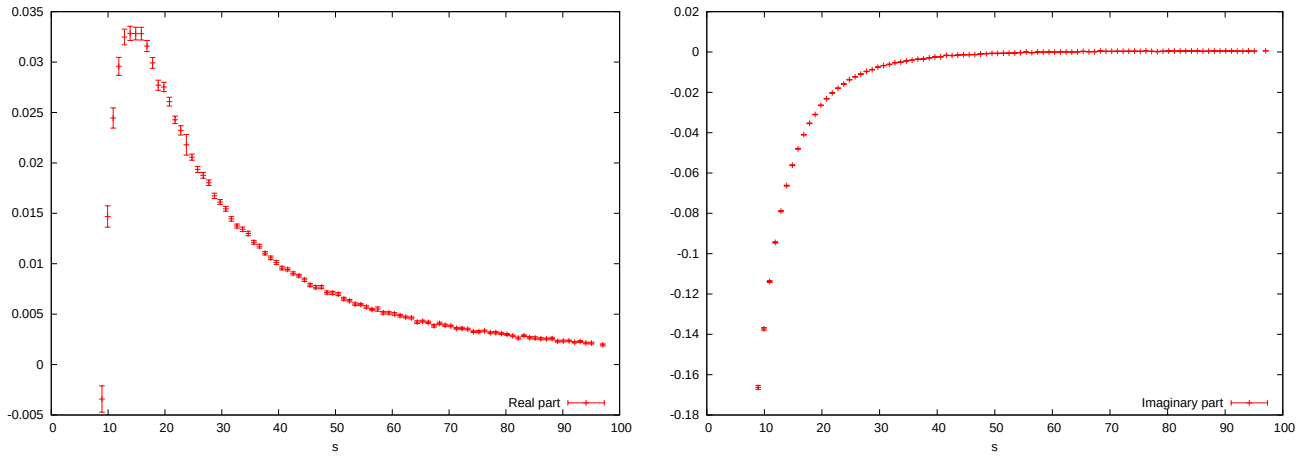

Figure 3. Real and imaginary parts of the finite part of the graph shown in Fig. 2, for the values $m_1^2 = 1, m_2^2 = 0.522, t = -3.978, u = -s - t + 2m_2^2$.

To define the scalar graph depicted in Fig. 2, the user can use the option `cutconstruct=1` as explained above. To this end, the vertices containing external legs need to be labelled with the same number as the corresponding external leg. Then the vertices containing only internal lines are labelled (in arbitrary order). The graph is defined in `mytemplate.m` by a list `proplist` containing the two vertex labels i_1, i_2 a certain propagator is connecting, and the mass m of that propagator, as $\{m, \{i_1, i_2\}\}$. For example, for the graph of Fig. 2

```
proplist={{0,{1,2}},{0,{1,6}},{0,{2,5}},{ms[1],{3,6}},{ms[1],{3,5}},
```

```
{ms[1],{4,5}},{ms[1],{4,6}}}; .
```

The kinematic conditions have to be given as

```
onshell={ssp[1]->0,ssp[2]->0,ssp[3]->ms[2],ssp[4]->ms[2]}; .
```

The expression `ssp[i]` is the standard name for p_i^2. Squared masses m_i^2 are denoted by `ms[i]`. The restriction to "standard names" will be lifted in the next release of SecDec, where the user will be able to define arbitrary symbols for the invariants.
Numerical results for this graph are shown in Fig. 3.

3.2. Momentum dependent two-loop corrections to neutral Higgs boson masses in the MSSM
In this section we briefly describe an application of SecDec where two-loop two-point integrals with up to four different mass scales are entering. Until recently, the $\mathcal{O}(\alpha_t\alpha_s)$ corrections to the neutral $\mathcal{CP}$-even Higgs boson masses in the MSSM have been known adopting a full $\overline{\text{DR}}$ scheme only [13, 14, 15], or neglecting the momentum dependence [16, 17, 18]. For a consistent inclusion of the corrections in the program FeynHiggs [16, 19, 20, 21, 22, 23], the top-quarks and top-squarks need to be renormalized on-shell. These momentum dependent $\mathcal{O}(\alpha_t\alpha_s)$ corrections missing so far in FeynHiggs were calculated in [24], where the 34 mass configurations of the analytically unknown two-loop two-point integrals are calculated numerically using SecDec. For further details we refer to [24, 25]. Very recently, an independent calculation of these corrections appeared [26], where the unknown integrals have been calculated numerically using the program `TSIL` [15], which is a dedicated program for the evaluation of two-loop two-point functions, based on differential equations.

At tree level, the mass matrix of the neutral $\mathcal{CP}$-even Higgs bosons in the (ϕ_1^0, ϕ_2^0) basis can be written as

$$M_{\text{Higgs}}^{2,\text{tree}} = \begin{pmatrix} M_A^2 \sin^2\beta + M_Z^2 \cos^2\beta & -(M_A^2 + M_Z^2)\sin\beta\cos\beta \\ -(M_A^2 + M_Z^2)\sin\beta\cos\beta & M_A^2\cos^2\beta + M_Z^2\sin^2\beta \end{pmatrix} , \tag{1}$$

where M_A is the mass of the $\mathcal{CP}$-odd neutral Higgs boson A. The rotation to the basis formed by the mass eigenstates H^0, h^0 is given by

$$\begin{pmatrix} H^0 \\ h^0 \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \phi_1^0 \\ \phi_2^0 \end{pmatrix} . \tag{2}$$

The higher-order corrected $\mathcal{CP}$-even Higgs boson masses in the MSSM are obtained from the corresponding propagators dressed by their self-energies. The inverse propagator matrix in the (ϕ_1^0, ϕ_2^0) basis is given by

$$(\Delta_{\text{Higgs}})^{-1} = -\text{i} \begin{pmatrix} p^2 - m_{\phi_1}^2 + \hat{\Sigma}_{\phi_1}(p^2) & -m_{\phi_1\phi_2}^2 + \hat{\Sigma}_{\phi_1\phi_2}(p^2) \\ -m_{\phi_1\phi_2}^2 + \hat{\Sigma}_{\phi_1\phi_2}(p^2) & p^2 - m_{\phi_2}^2 + \hat{\Sigma}_{\phi_2}(p^2) \end{pmatrix} , \tag{3}$$

where the $\hat{\Sigma}(p^2)$ denote the momentum-dependent renormalized Higgs-boson self-energies, p being the external momentum. The latter have been calculated at the two-loop level, at order $\alpha_s\alpha_t$.

Our calculation is performed in the Feynman-diagrammatic approach. We adopt a hybrid on-shell/$\overline{\text{DR}}$ scheme, in line with the renormalization of previous higher-order contributions included in the program FeynHiggs[4], see Ref. [24] for more details. To obtain expressions for the unrenormalized self-energies and tadpoles at $\mathcal{O}(\alpha_t\alpha_s)$, the evaluation of genuine two-loop diagrams and one-loop graphs with counterterm insertions is required. Example diagrams for the neutral Higgs-boson self-energies are shown in Fig. 4.

[4] We neglect a numerically insignificant shift in the value of $\tan\beta$.

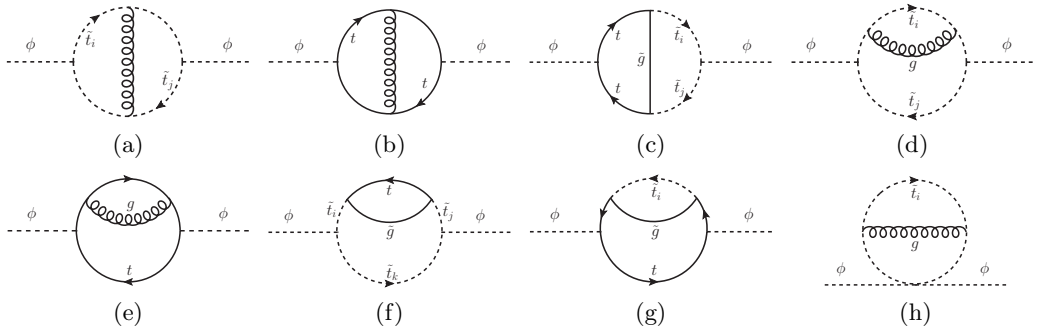

Figure 4. Examples of two-loop diagrams enetring the Higgs-boson self-energies ($\phi = h, H, A$).

For the counterterm-insertions, one-loop diagrams with external top quarks/squarks have to be evaluated as well. The complete set of contributing Feynman diagrams has been generated with the program `FeynArts` [27, 28] (using the model file including counterterms from Ref. [29]). Tensor reduction and the evaluation of traces was done with support from the programs `FormCalc` [30] and `TwoCalc` [31]. The resulting two-loop integrals which depend on the external momentum contain four topologies for which only partial analytical results are available. These topologies are shown in Fig. 5. They occur in 34 different mass configurations, and have been evaluated with SecDec.

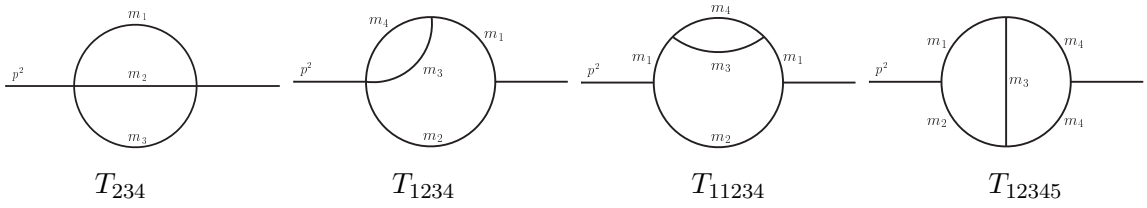

Figure 5. Topologies which have been calculated numerically using SecDec.

Computation of mass shifts
The calculation of the self-energies is performed in the (ϕ_1^0, ϕ_2^0) basis. To be consistent with all other higher-order contributions to the Higgs-boson masses incorporated in the program FeynHiggs, the renormalized self-energies in the (ϕ_1^0, ϕ_2^0) basis is rotated into the physical (h^0, H^0) basis, where the tree-level propagator matrix is diagonal, via

$$\hat{\Sigma}^{(2)}_{H^0H^0} = \cos^2\alpha\,\hat{\Sigma}^{(2)}_{\phi_1^0\phi_1^0} + \sin^2\alpha\,\hat{\Sigma}^{(2)}_{\phi_2^0\phi_2^0} + \sin(2\alpha)\,\hat{\Sigma}^{(2)}_{\phi_1^0\phi_2^0}\,, \tag{4a}$$

$$\hat{\Sigma}^{(2)}_{h^0h^0} = \sin^2\alpha\,\hat{\Sigma}^{(2)}_{\phi_1^0\phi_1^0} + \cos^2\alpha\,\hat{\Sigma}^{(2)}_{\phi_2^0\phi_2^0} - \sin(2\alpha)\,\hat{\Sigma}^{(2)}_{\phi_1^0\phi_2^0}\,, \tag{4b}$$

$$\hat{\Sigma}^{(2)}_{h^0H^0} = \sin\alpha\cos\alpha\,(\hat{\Sigma}^{(2)}_{\phi_2^0\phi_2^0} - \hat{\Sigma}^{(2)}_{\phi_1^0\phi_1^0}) + \cos(2\alpha)\,\hat{\Sigma}^{(2)}_{\phi_1^0\phi_2^0}\,, \tag{4c}$$

and α is the tree-level mixing angle.

The resulting new contributions to the neutral $\mathcal{CP}$-even Higgs-boson self-energies, containing all momentum-dependent and additional constant terms, are assigned to the differences

$$\Delta\hat{\Sigma}^{(2)}_{ab}(p^2) = \hat{\Sigma}^{(2)}_{ab}(p^2) - \tilde{\Sigma}^{(2)}_{ab}(0)\,, \qquad ab = \{H^0H^0, h^0H^0, h^0h^0\}\,. \tag{5}$$

Note the tilde (not hat) on $\tilde{\Sigma}^{(2)}(0)$, which signifies that not only the self-energies are evaluated at zero external momentum but also the corresponding counter-terms, following Refs. [32, 33, 16].

A finite shift $\Delta\hat{\Sigma}^{(2)}(0)$ therefore remains in the limit $p^2 \to 0$ due to $\delta m^{2(2)}_{A^0} = \mathrm{Re}\,\Sigma^{(2)}_{A^0A^0}(m^2_{A^0})$ being computed at $p^2 = m^2_{A^0}$ in $\hat{\Sigma}^{(2)}$, but at $p^2 = 0$ in $\tilde{\Sigma}^{(2)}$.

According to Eq. (3), the $\mathcal{CP}$-even Higgs boson masses are determined by the poles of the h^0-H^0-propagator matrix. This is equivalent to solving the equation

$$\left[p^2 - m^2_{h^0} + \hat{\Sigma}_{h^0h^0}(p^2)\right]\left[p^2 - m^2_{H^0} + \hat{\Sigma}_{H^0H^0}(p^2)\right] - \left[\hat{\Sigma}_{h^0H^0}(p^2)\right]^2 = 0 \ , \tag{6}$$

yielding the loop-corrected pole masses, M_h and M_H.

Numerical results
In our numerical analysis we find that the effects on the light $\mathcal{CP}$-even Higgs boson mass, M_h, are sensitive to the value of the gluino mass, $m_{\tilde{g}}$. For values of $m_{\tilde{g}} \sim 1.5$ TeV corrections to M_h of about -50 MeV are found, at the level of the anticipated future ILC accuracy. For very large gluino masses, $m_{\tilde{g}} \gtrsim 4$ TeV, on the other hand, substantially larger corrections are found, at the level of the current experimental accuracy of ~ 500 MeV. The new results of $\mathcal{O}(\alpha_t\alpha_s)$ including momentum dependence have been implemented into the program FeynHiggs.

4. Conclusions

We have described the program SecDec, with emphasis on useful features and new applications. SecDec is a flexible tool which can be used to factorise poles from dimensionally regulated parameter integrals, and to evaluate the pole coefficients numerically. As there is no restriction on the kinematic invariants, the program is particularly useful for two-loop integrals with several mass scales, where analytic approaches have difficulties.

The program further contains scripts to facilitate scans over large ranges of numerical parameters, based on functions for the Laurent series expansion of the integral which can be produced once and for all.

The forthcoming version 3.0 of the program will contain further improvements, in particular offer new decomposition strategies.

Acknowledgments

We would like to thank the organizers of ACAT2014 for the nice conference. We also thank Thomas Hahn, Sven Heinemeyer and Wolfgang Hollik for collaboration on the MSSM Higgs project, and Johannes Schlenk for his contributions towards SecDec version 3.0.

References

[1] Carter J and Heinrich G 2011 *Comput.Phys.Commun.* **182** 1566–1581 (*Preprint* 1011.5493)
[2] Borowka S, Carter J and Heinrich G 2013 *Comput.Phys.Commun.* **184** 396–408 (*Preprint* 1204.4152)
[3] Borowka S and Heinrich G 2013 *Comput.Phys.Commun.* **184** 2552–2561 (*Preprint* 1303.1157)
[4] Binoth T and Heinrich G 2000 *Nucl. Phys.* **B585** 741–759 (*Preprint* hep-ph/0004013)
[5] Heinrich G 2008 *Int. J. Mod. Phys.* **A23** 1457–1486 (*Preprint* 0803.4177)
[6] Bogner C and Weinzierl S 2008 *Comput. Phys. Commun.* **178** 596–610 (*Preprint* 0709.4092)
[7] Gluza J, Kajda K, Riemann T and Yundin V 2011 *Eur.Phys.J.* **C71** 1516 (*Preprint* 1010.1667)
[8] Smirnov A and Tentyukov M 2009 *Comput.Phys.Commun.* **180** 735–746 (*Preprint* 0807.4129)
[9] Smirnov A V 2014 *Comput.Phys.Commun.* **185** 2090–2100 (*Preprint* 1312.3186)
[10] Hahn T 2005 *Comput. Phys. Commun.* **168** 78–95 (*Preprint* hep-ph/0404043)
[11] Agrawal S, Hahn T and Mirabella E 2011 (*Preprint* 1112.0124)
[12] Kawabata S 1995 *Comp. Phys. Commun.* **88** 309–326
[13] Martin S P 2004 *Phys.Rev.* **D70** 016005 (*Preprint* hep-ph/0312092)
[14] Martin S P 2005 *Phys.Rev.* **D71** 016012 (*Preprint* hep-ph/0405022)
[15] Martin S P and Robertson D G 2006 *Comput.Phys.Commun.* **174** 133–151 (*Preprint* hep-ph/0501132)
[16] Heinemeyer S, Hollik W and Weiglein G 1999 *Eur.Phys.J.* **C9** 343–366 (*Preprint* hep-ph/9812472)

[17] Heinemeyer S, Hollik W and Weiglein G 1999 *Phys.Lett.* **B455** 179–191 (*Preprint* hep-ph/9903404)
[18] Degrassi G, Slavich P and Zwirner F 2001 *Nucl.Phys.* **B611** 403–422 (*Preprint* hep-ph/0105096)
[19] Heinemeyer S, Hollik W and Weiglein G 2000 *Comput.Phys.Commun.* **124** 76–89 (*Preprint* hep-ph/9812320)
[20] Hahn T, Heinemeyer S, Hollik W, Rzehak H and Weiglein G 2009 *Comput.Phys.Commun.* **180** 1426–1427
[21] Hahn T, Heinemeyer S, Hollik W, Rzehak H and Weiglein G 2014 *Phys.Rev.Lett.* **112** 141801 (*Preprint* 1312.4937)
[22] Degrassi G, Heinemeyer S, Hollik W, Slavich P and Weiglein G 2003 *Eur.Phys.J.* **C28** 133–143 (*Preprint* hep-ph/0212020)
[23] Frank M, Hahn T, Heinemeyer S, Hollik W, Rzehak H *et al.* 2007 *JHEP* **0702** 047 (*Preprint* hep-ph/0611326)
[24] Borowka S, Hahn T, Heinemeyer S, Heinrich G and Hollik W 2014 *Eur.Phys.J.* **C74** 2994 (*Preprint* 1404.7074)
[25] Borowka S 2014 *Ph.D. thesis, Max Planck Institute for Physics/Technical University Munich* (*Preprint* 1410.7939)
[26] Degrassi G, Di Vita S and Slavich P 2014 (*Preprint* 1410.3432)
[27] Hahn T 2001 *Comput.Phys.Commun.* **140** 418–431 (*Preprint* hep-ph/0012260)
[28] Hahn T and Schappacher C 2002 *Comput.Phys.Commun.* **143** 54–68 (*Preprint* hep-ph/0105349)
[29] Fritzsche T, Hahn T, Heinemeyer S, von der Pahlen F, Rzehak H *et al.* 2014 *Comput.Phys.Commun.* **185** 1529–1545 (*Preprint* 1309.1692)
[30] Hahn T and Perez-Victoria M 1999 *Comput.Phys.Commun.* **118** 153–165 (*Preprint* hep-ph/9807565)
[31] Weiglein G, Scharf R and Böhm M 1994 *Nucl.Phys.* **B416** 606–644 (*Preprint* hep-ph/9310358)
[32] Heinemeyer S, Hollik W and Weiglein G 1998 *Phys.Rev.* **D58** 091701 (*Preprint* hep-ph/9803277)
[33] Heinemeyer S, Hollik W and Weiglein G 1998 *Phys.Lett.* **B440** 296–304 (*Preprint* hep-ph/9807423)